

Index

A

- a.out file (dbx), 3-2
- _ADAWI built-in function, 10-4
- Additive operators, 6-10
- Address correlation table
 - effect of debug argument (-V option), 2-11
- Address-of operator, 6-8, 7-8
- Aggregates, 7-11 to 7-22
 - arrays, 7-11
 - See also Bracket operators ([])
 - defined, 7-2, 7-11
 - introduction to, 4-14
 - structures, 7-11, 7-15
 - unions, 7-11, 7-15
 - variant, 7-20
- alias command (dbx), 3-7
- _align modifier, 8-18
- Allocation
 - modifiers, 8-16
- AND bitwise operator, 6-12
- argc
 - main function argument, 4-28
- Arguments, 4-26 to 4-27
 - array identifiers used as, 4-27
 - command-line, 4-28
 - conversion of, 4-26
 - function prototypes, 1-6, 4-24
 - functions used as, 4-27
 - in #define preprocessor macros, 9-3
 - introduction to, 4-3
 - rules governing, 4-26
 - to a function
 - conversion of, 4-26, 6-2, 6-17
- argv
 - main function argument, 4-28
- Arithmetic conversion rules, 6-16
- Arithmetic operators
 - negation, 6-7
- Array expressions
 - syntax for dbx, 3-4
- Arrays, 7-11
 - as expressions, 6-3
 - declaration of, 7-11
 - initialization of, 7-13
 - introduction to, 4-14
 - references to, 6-3
- asm pseudo function call, 1-9
- assign command (dbx), 3-7
- Assignment
 - operators, 6-14
 - precedence of, 6-6, 6-14

- Asterisk operator (*), 7-8
- [auto] specifier, 8-4, 8-8 to 8-9

B

- \b (backspace), 7-5
- _BBCCI built-in function, 10-4
- _BBSSI built-in function, 10-4
- Binary operators
 - additive, 6-10
 - bitwise, 6-12
 - equality, 6-11
 - logical, 6-13
 - multiplication, 6-11
 - precedence of, 6-6
 - relational, 6-11
 - shift, 6-13
- Bit fields, 7-22
- Bitwise operators, 6-12
- Blocks, 4-31 to 4-32, 5-2 to 5-3
- Boolean algebra, 6-12
 - See also Bitwise operators
- b option
 - vcc command option, 2-8
- B option
 - vcc command option, 2-8
- Braces ({})
 - in compound statements, 4-31
 - in initializer lists, 7-13
- Bracket operators ([]), 6-3
- break statement, 5-8
 - use with switch statement, 5-4
- bss program section, C-5
- Built-In functions, 10-3 to 10-15
 - _ADAWI, 10-4
 - _BBCCI, 10-4
 - _BBSSI, 10-4
 - _FFC, 10-5
 - _FFS, 10-5
 - _HALT, 10-6
 - _INSQHI, 10-6
 - _INSQTI, 10-6
 - _INSQUE, 10-7
 - _LDPCTX, 10-7
 - _LOCC, 10-7
 - _MFPR, 10-8
 - _MOVC3, 10-8
 - _MOVC5, 10-9
 - _MOVPSL, 10-10
 - _MTPR, 10-10
 - _PROBER, 10-10

Built-In functions (Cont.)

- `_PROBEW`, 10-11
- `_READ_GPR`, 10-11
- `_REMQHI`, 10-11
- `_REMQTI`, 10-12
- `_REMQUE`, 10-12
- `_SCANC`, 10-13
- `_SIMPLE_READ`, 10-13
- `_SIMPLE_WRITE`, 10-14
- `_SKPC`, 10-14
- `_SPANC`, 10-15
- `_SVPCTX`, 10-15
- `_WRITE_GPR`, 10-15

builtins pragma, 9-9

C

call command (dbx), 3-8

case label, 5-4

Case sensitivity, 4-29

Cast operator, 6-9

catch command (dbx), 3-8, 3-9

C compiler

See Compiler

Character

constants, 7-5

data type

variable, 7-3

strings, 7-11, 7-14

See also Arrays

introduction to, 4-14

Character-string constants, 7-14

See also Arrays

limit of length, 7-14

C language

See also VAX C language

introduction to, 4-1 to 4-2

Command-line arguments, 4-28

conversion of, 4-29

Command-line retention (dbx), 3-3

Comma operator, 6-15

precedence of, 6-6

Comments, 4-32

Compilation options

options affecting debugging, 3-2

options affecting output

contents of source listing file (show), 2-15

debugging information (debug and optimize),
2-7, 2-14

messages (standard and warnings), 2-16 to
2-17

object code listing (machine_code), 2-14

object files (lobject and object), 2-13, 2-14

source listing (list), 2-13

options affecting processing

G_float versus D_float, 2-12

optimization (optimize), 2-14

Compilation unit

in determining scope, 8-2

Compiler

diagnostic messages issued by
general description, 2-17

functions, 2-4

input to linker, 2-4

phases, 1-2

Compiler differences

Compiler differences (Cont.)

pcc and VAX C, 1-2 to 1-10

VAX C/VMS and VAX C/ULTRIX, 1-10 to 1-12

Compound statements, 4-31, 5-2

\$\$COMSYMS, A-7

Conditional compilation, 9-5 to 9-6

Conditional operator, 6-13

precedence of, 6-6

Conditional operators (C)

use in debugging, 3-3

Conditional statements, 5-3 to 5-6

const modifier, 8-16

Constants, 7-1

character, 7-5

escape sequence, 7-5

hexadecimal escape sequence, 7-6

character strings, 7-14

floating-point, 7-7

identifier, 9-3

integer, 7-4

values of, 7-1

cont command (dbx), 3-8

Continuation

strings, 7-14

continue statement, 5-8

Control flow statements, 5-1 to 5-2

Conversion rules, D-4

Conversions, 6-15 to 6-17

arithmetic, 6-15, 6-16

of data types, 6-15

of function arguments, 6-2, 6-17

with cast operator, 6-9

-c option

vcc command option, 2-8

Copy commands

DECnet-ULTRIX dcp command, C-1

DECnet-VAX COPY command, C-2

Cross-reference information

compiling

cross_reference argument, 2-10

in output listing

storage map section, 2-21

cross_reference argument

vcc command (-V option), 2-10

cross_reference option

effect on output listing, 2-21

D

Data definitions

external, 8-13

scope of external, 8-13

Data files

See Files

Data structures, 7-11

See also Aggregates

Data-type keywords, D-2

Data-type modifiers, 8-16, D-3

Data types, 7-1 to 7-23

conversion of, 6-15

function prototypes, 1-6, 4-24

introduction to, 4-3

modifiers, 8-16

supported by dbx debugger, 3-10

__DATE__ predefined macro, 10-2

dbx command, 3-2

dbx commands

dbx commands (Cont.)

- data types displayed by, 3-10
- summary, 3-5 to 3-7

dbx conventions

- array expression syntax, 3-4
- initialization files, 3-3
- use of conditional operators, 3-3

dcp command (DECnet-ULTRIX), C-1

debug argument

- vcc command (-V option), 2-10 to 2-11

Debugger

- example sessions, 3-15 to 3-17

Debugging operations

- assign values (assign), 3-7
- assign value to debugger variable (set), 3-11
- change command names (alias), 3-7
- change function level (up), 3-13
- change functions (func), 3-9
- change search directories (use), 3-13
- change source files (file), 3-9
- delete debugger variable (unset), 3-12
- display declarations (whatis), 3-13
- display help information (help), 3-9
- display memory locations
 - by means of addresses, 3-13
- display names and values (dump), 3-8
- display source lines (list), 3-9
- display symbol qualifications (whereis), 3-13
- display symbol qualifications (which), 3-13
- display traces and stops (status), 3-11
- display values of expressions (print), 3-10
- edit a source file (edit), 3-8
- enable tracing operations
 - machine instruction level (tracei), 3-14
 - source line level (trace), 3-12
- enter shell commands (sh), 3-11
- execute a program (run/rerun), 3-10, 3-11
- execute dbx commands
 - in specified files (source), 3-11
- execute to next line (next), 3-9
- execute to next line (step), 3-11
- execute to next machine instruction (nexti), 3-14
- execute to next machine instruction (stepi), 3-14
- exit the debugger (quit), 3-10
- handle interrupt signals (catch and ignore), 3-8, 3-9
- invoke other functions (call), 3-8
- invoking dbx, 3-2
- list active functions (where), 3-13
- remove alias (unalias), 3-12
- remove traces and stops (delete), 3-8
- restart execution (cont), 3-8
- search backward (?), 3-13
- search forward (/), 3-13
- stop execution
 - machine instruction debugging (stopi), 3-14
 - source line debugging (stop), 3-12
- stop execution at return (return), 3-10
- trace execution (trace), 3-12
- with optimized programs, 3-4

Declarations, 7-1 to 7-3

aggregate

- arrays, 7-11
- structures, 7-15
- unions, 7-15

format of, 7-2

function

Declarations

function (Cont.)

- void, 7-22

function prototypes, 1-6, 4-24

inside of blocks, 5-3

interpreting, 7-23 to 7-27

overlapping scope of, 4-31

parameters, 4-27

position of

- determining scope, 8-2 to 8-3

scalar

- character constant, 7-5

- character variable, 7-3

- enumerated, 7-9

- integer, 7-3

- pointer, 7-8

vacuous tag declarations, 7-17

DECnet-ULTRIX dcp command, C-1

DECnet-VAX COPY command, C-2

Decrement operator, 6-7

- side effects within macros, 9-4

default label, 5-4

Default options, 1-4

#define directive, 2-11, 9-1, 9-5

defined operator, 9-7

define argument

- vcc command (-V option), 2-11 to 2-12

Definitions, 7-1

function

- void, 7-22

functions, 4-18

delete command (dbx), 3-8

Dereferencing, 6-8

See also Pointers

Diagnostic messages

- from the linker, B-33 to B-43

- general description, 2-18

- from the VAX C compiler, B-2 to B-33

- general description, 2-17

- from the vcc command, B-1 to B-2

Differences

compiler

- pcc and VAX C, 1-2 to 1-10

- VAX C/VMS and VAX C/ULTRIX, 1-10 to 1-12

Directives

- #define, 9-1

- #elif, 9-5

- #else, 9-5

- #endif, 9-5

- #if, 9-5

- #ifdef, 9-5

- #ifndef, 9-5

- #include, 9-7

- #line, 9-8

- #pragma, 9-9 to 9-12

- #undef, 9-5

Division operator, 6-11

do statement, 5-7

-D option

- vcc command option, 2-8

double keyword, 7-7

down command (dbx), 3-8

dump command (dbx), 3-8

D_float data implementation

- effect of g_float argument

- vcc command (-V option), 2-12

D_floating representation, 7-7

E

edit command (dbx), 3-8

Editor

See Text editor

#elif directive, 9-5

#else directive, 9-5

-Em option, 1-3

vcc command option, 2-8

#endif directive, 9-5

Entry point

main, 2-18

enum keyword, 7-9 to 7-11

Enumerated data type, 7-9 to 7-11

declaration of, 7-9

envp

main function argument, 4-28

-E option, 1-2

vcc command option, 2-8

Equality operators, 6-11

Error (severity)

meaning to linker, 2-18

Error messages

from the linker, B-33 to B-43

general description, 2-18

from the VAX C compiler, B-2 to B-33

general description, 2-17

from the vcc command, B-1 to B-2

Escape sequences, 7-5, D-5

hexadecimal values, 7-6

Evaluating expressions

See Expressions

EXCLUSIVE OR bitwise operator, 6-12

Executable image sizes

VAX C on ULTRIX, C-4

Expressions, 5-2 to 5-3, 6-1 to 6-17

assignment, 6-14

as statements, 5-2

binary

additive, 6-10

bitwise, 6-12

equality, 6-11

logical, 6-13

multiplication, 6-11

relational, 6-11

shift, 6-13

comma, 6-15

conditional, 6-13

evaluation order

ambiguity of, 6-8

primary, 6-2 to 6-4

array reference, 6-3

formal syntax of, 6-2

function call, 6-2

lvalues, 6-1

parentheses, 6-2

rvalues, 6-1

structure reference, 6-4

union reference, 6-4

unary

addressed, 6-8

cast, 6-9

increment and decrement, 6-7

negation, 6-7

Expressions

unary (Cont.)

one's complement, 6-9

sizeof, 6-10

[extern] specifier, 8-4, 8-10, 8-11

External storage class, 8-10 to 8-11

compared to global, 8-13 to 8-14

data definitions, 8-13

F

\f (form feed), 7-5

Fatal (severity)

meaning to linker, 2-18

_FFC built-in function, 10-5

_FFS built-in function, 10-5

File

creating source files, 2-3

editing files, 2-3

__FILE__ predefined macro, 10-2

file command (dbx), 3-9

Files

data files

copying between VMS and ULTRIX, C-5

magnetic tape files

handling with DEC/Shell tar utility, C-2

program files

copying between VMS and ULTRIX, C-1 to C-3

float

in prototype declarations, 4-25

float keyword, 7-6

Floating-point

constants, 7-7

data type

declaration of, 7-6

double, 7-7

D_floating, 7-7

float, 7-7

F_floating, 7-7

G_floating, 7-7

long, 7-7

long float, 7-7

precision of, 7-6

sizes of, 7-6

-f option

vcc command option, 2-8

for statement, 5-6, 5-7

introduction to, 4-10

\$fortmsgfile, C-4

Forward referencing

structures, 7-17

func command (dbx), 3-9

Function

built-in, 10-3 to 10-15

Function definition

arguments

conversion of, 6-17

Functions

address of, 4-27, 6-3

as arguments, 4-27

calls to, 6-2

within macros, 9-4

definitions of, 4-18 to 4-27

argument conversion, 6-2

arguments, 4-19, 4-26

body, 4-19

Functions

definitions of (Cont.)

- main function, 4-20
- main_program option, 4-20
- names of, 4-20
- parameters, 4-19, 4-26
- introduction to, 4-3
- prototypes, 1-6, 4-24
- return values of, 4-23
- scope of, 4-20
- undeclared, 6-2
- void keyword, 7-22

G

Generic pointers, 7-22

globaldef specifier, 8-11, 8-13

globaldef data type

- with enumerated values, 8-15

globalref specifier, 8-11, 8-13

globalref data type

- See also Storage classes
- with enumerated values, 8-15

Global storage class, 8-11 to 8-15

- compared to external, 8-13 to 8-14
- variable initialization, 8-11

globalvalue specifier, 8-14

-g option

- vcc command option, 2-8

-g option (vcc command)

- effect on debugging, 3-2

goto statement, 5-2

g_float argument

- vcc command (-V option), 2-12

G_float data implementation

- g_float argument (-V option), 2-12

G_floating representation, 7-7

H

_HALT built-in function, 10-6

help command (dbx), 3-9

I

Identifiers, 4-29

if statement, 5-3

- introduction to, 4-7

#if directive, 9-5

#if defined operator, 9-7

#ifdef directive, 9-5

#ifndef directive, 9-5

ignore command (dbx), 3-8, 3-9

Image sizes

- VAX C on ULTRIX, C-4

#include directive, 9-7

Increment operator, 6-7

- side effects within macros, 9-4

Indirection operator, 7-8

Initialization

- arrays, 7-11
- characters, 7-3
- character-string variables, 7-14
- integers, 7-3
- of global variables, 8-11
- structures, 7-18

Initialization files (dbx), 3-3

inline pragma, 9-9

Input and output (I/O)

- introduction to, 4-4

_INSQHI built-in function, 10-6

_INSQTI built-in function, 10-6

_INSQUE built-in function, 10-7

Integer constants, 7-4

- invalid, 7-5

Integer data types

- declaration of, 7-3

- sizes of, 7-3

Internal storage class, 8-7 to 8-9

Interrupting statements, 5-8 to 5-9

Interrupt signals

- handling in dbx

- catch and ignore, 3-8, 3-9

-I option

- vcc command option, 2-8

K

Keywords

- _align, 8-18

- auto, 8-4, 8-8

- break, 5-8

- case, 5-4

- const, 8-16

- continue, 5-8

- default, 5-4

- do, 5-7

- else, 5-3

- enum**, 7-9 to 7-11

- extern, 8-4, 8-10, 8-11

- for, 5-6

- globaldef**, 8-11

- globalref**, 8-11

- globalvalue**, 8-14

- goto, 5-2

- if, 5-3

- introduction to, 4-4

- list of, 4-29 to 4-31

- noshare, 8-18

- readonly, 8-18

- register, 8-9

- return, 5-9

- sizeof, 6-10

- static, 8-9

- struct, 7-15

- switch, 5-4

- union, 7-15

- void**, 7-9

- volatile, 8-17

- while, 5-7

-K option

- vcc command option, 2-8

L

Labeled statements, 5-2

_LDPCTX built-in function, 10-7

Lexical scope, 8-3 to 8-4

Lifetime

- of stored objects, 8-7

#line directives, 9-8

__LINE__ predefined macro, 10-3

Linker

- command options, A-2 to A-3

Linker (Cont.)

- diagnostic messages issued by
 - general description, 2-18
 - effect on files, A-1
 - lk command, A-1
 - vcc command options, A-1
- Link-time scope, 8-3 to 8-4
- lint utility, 1-4, 4-32
 - function prototypes, 1-6, 4-24
- list command (dbx), 3-9
- Listing output, 1-4
- list option
 - vcc command (-V option), 2-13
- lk command
 - See Linker, A-1
- lk Linker Image Map, 2-23 to 2-30
- _LOCC built-in function, 10-7
- Logical
 - negation operator, 6-7
 - operators, 6-13
- long** keyword, 7-3, 7-7
- Looping statements, 5-6 to 5-8
 - See also Statements
 - introduction to, 4-9
- l option
 - vcc command option, 2-8
- lvalues, 6-1
 - introduction to, 4-12

M

- Machine code output listing
 - general description, 2-19 to 2-20
- machine_code argument
 - vcc command (-V option), 2-14
- Macro
 - substitution
 - in #include directives, 9-8
- Macro definitions, 9-3 to 9-5
 - canceling, 9-5
 - listing substituted lines, 9-5
 - naming parameters in, 9-4
 - on command line, 2-11
 - possible side effects, 9-4
- Macros
 - predefined
 - __DATE__, 10-2
 - __FILE__, 10-2
 - __LINE__, 10-3
 - __TIME__, 10-3
 - system-identification, 10-1
- Macro substitutions, 4-5, 9-3 to 9-5
- Magnetic tape files
 - See Files
- Main function, 4-20
 - syntax of, 4-28
 - with main_program option, 4-20
- Md option
 - vcc command option, 2-8
- Members
 - defined, 7-2
 - variant aggregates, 7-20
- member_alignment** pragma, 9-11
- Messages
 - from the linker, B-33 to B-43
 - from the VAX C compiler, B-2 to B-33

Messages (Cont.)

- from the vcc command, B-1 to B-2
- issued by compiler
 - general description, 2-17
- issued by linker
 - general description, 2-18
- Message severity
 - E (error)
 - meaning to compiler, B-33
 - F (fatal)
 - meaning to compiler, B-33
 - I (informational)
 - meaning to compiler, B-33
 - W (warning)
 - meaning to compiler, B-33
- _MFPR built-in function, 10-8
- Mg option
 - vcc command option, 2-8
- Migrating source programs
 - See Portability concerns
- Modifiers
 - storage class, 8-17
- Modulo operator, 6-11
- _MOV3 built-in function, 10-8
- _MOV5 built-in function, 10-9
- _MOVPSL built-in function, 10-10
- _MTPR built-in function, 10-10
- Multiplication operators, 6-11

N

- \n (newline), 7-5
- Negation
 - arithmetic and logical, 6-7
 - unsigned, 6-7
- next command (dbx), 3-9
- nexti command (dbx), 3-14
- noinline** pragma, 9-9
- nomember_alignment** pragma, 9-11
- noshare** modifier, 8-18
- nostandard** pragma, 9-11
- NUL character, 4-15
- Null
 - pointer, 7-8
- Null statement, 5-1

O

- object argument
 - vcc command (-V option), 2-14
- Object file formats, 1-3
- Object module
 - in determining scope, 8-2
- Objects
 - of variables, 7-1
- Octal constants, 7-4
- .o file type, 2-7
- om utility, B-1
- One's complement operator, 6-9
- o option
 - vcc command option, 2-8
- O option
 - vcc command option, 2-8
- Operand conversion, 6-16
- Operators, 6-4 to 6-15
 - assignment, 6-14 to 6-15
 - ambiguity of, 6-15

Operators (Cont.)

- binary, 6-10 to 6-13
 - additive, 6-10
 - bitwise, 6-12
 - equality, 6-11
 - logical, 6-13
 - multiplication, 6-11
 - relational, 6-11
 - shift, 6-13
 - bracket ([]), 6-3
 - categories of, 6-5
 - comma, 6-15
 - conditional, 6-13
 - defined, 9-7
 - list of, 6-4
 - precedence of, 6-6, D-4
 - unary, 6-7 to 6-10
 - address of, 6-8
 - cast, 6-9
 - increment and decrement, 6-7
 - indirection, 6-8
 - negation, 6-7
 - one's complement, 6-9
- Optimization capabilities, 1-3
- OR bitwise operator, 6-12
- Output listing, 2-18 to 2-23
- machine code section, 2-19 to 2-20
 - options affecting output
 - See Compilation options
 - storage map section, 2-20 to 2-23

P

- Parameters, 4-26 to 4-27
- declaration of, 4-19, 4-27
 - function prototypes, 1-6, 4-24
 - in **#define** preprocessor macros, 9-3
 - introduction to, 4-3
 - main function, 4-28
 - rules governing, 4-26
- pcc source programs
- portability concerns, 1-1
- Period operator, 6-4
- pg option
- vcc command option, 2-8
- Pointers
- declaration of, 7-8
 - generic, 1-7, 7-22
 - introduction to, 4-12
 - null, 7-8
 - unary operator, 6-8
 - void**, 7-9
- p option
- vcc command option, 2-8
- Portability concerns, 1-1 to 1-12, 4-1 to 4-2
- See also C language
 - char *** generic-pointer notation, 7-9
 - character-string constants, 7-5
 - character string length, 7-14
 - comparing pointers and integers, 6-11
 - deviations assignment operators, 6-15
 - direction of bit field packing, 7-22
 - global storage classes, 8-13
 - global system status values, 8-14
 - int values on a VAX system, 7-4
 - length of argument list, 4-26

Portability concerns (Cont.)

- length of bit fields, 7-22
 - length of identifiers, 4-29
 - lexical scope and compilation units, 8-2
 - long float keyword, 7-7
 - main_program option, 4-20
 - octal constants, 7-4
 - parameter declarations, 4-27
 - predefined symbols, 4-31
 - predefined system-definition macros, 10-1
 - preprocessor implementations, 9-1
 - preprocessor substitutions, 10-1, 10-3
 - referencing aggregate members, 7-18
 - structure alignment, 7-15
- #pragma** preprocessor directive, 1-7
- #pragma** directive, 9-9 to 9-12
- Pragmas
- builtins**, 9-9, 10-3
 - inline**, 9-9
 - member_alignment, 7-15, 9-11
 - standard**, 9-11
- Precedence of operators, 6-6
- in interpreting declarations, 7-24
- Predefined macros, 10-1 to 10-3
- Predefined symbols, 4-31, 10-3
- Preprocessor (cpp), 1-2
- Preprocessor directives, 9-1 to 9-12, D-5
- #define**, 9-1
 - #elif**, 9-5
 - #else**, 9-5
 - #endif**, 9-5
 - #if**, 9-5
 - #ifdef**, 9-5
 - #ifndef**, 9-5
 - #include**, 9-7
 - macro substitution, 9-8
 - #line**, 9-8
 - #pragma**, 9-9 to 9-12
 - #undef**, 9-5
- Preprocessor substitutions, 10-1, 10-3
- Primary expressions, 6-2 to 6-4
- See also Expressions
 - array reference, 6-3
 - function call, 6-2
 - lvalues, 6-1
 - parentheses, 6-2
 - rvalues, 6-1
 - structure reference, 6-4
 - union reference, 6-4
- Primary operators
- precedence of, 6-6
- print command (dbx), 3-10
- Privacy, 8-6
- See also Scope
- _PROBER** built-in function, 10-10
- _PROBEW** built-in function, 10-11
- Program files
- See Files
- Programs
- compiling, 2-1
- Program section (psect)
- attributes, A-4
 - defined, 8-7
 - for global symbols, 8-11
 - in linker map, 2-25
- Program structure, 4-18 to 4-33

Prototypes, 1–6
 function, 4–24
PSL, 10–10

Q

quit command (dbx), 3–10

R

\r (carriage return), 7–5
_READ_GPR built-in function, 10–11
readonly modifier, 8–18
References
 forward, 1–3
 unresolved, 2–18
register specifier, 6–8, 7–8, 8–9
Relational operators, 6–11
_REMQHI built-in function, 10–11
_REMQTI built-in function, 10–12
_REMQUE built-in function, 10–12
rerun command (dbx), 3–10, 3–11
Reserved words, 4–29 to 4–31
return statement, 4–19, 5–9
return command (dbx), 3–10
Right-arrow operator, 6–4
run command (dbx), 3–11
rvalues, 6–1
 introduction to, 4–12

S

Scalar data types, 7–3 to 7–11
 declarations, 7–3
 character, 7–3
 enumerated, 7–9
 floating-point, 7–6
 integer, 7–3
 pointers, 7–8
 defined, 7–2
 introduction to, 4–14
_SCANC built-in function, 10–13
Scope, 8–1 to 8–6
 auto variables, 4–31
 in a compilation unit, 8–2
 in an object module, 8–2
 in a program, 8–2
 lexical scope, 8–3
 link-time scope, 8–3
 of external data, 8–13
 of functions, 4–20
 position of declarations, 8–2 to 8–3
set command (dbx), 3–11
sh command (dbx), 3–11
Shell commands
 issuing in debugger (sh), 3–11
Shift operators, 6–13
show argument
 vcc command (–V option), 2–15
Signals
 handling in dbx
 catch and ignore, 3–8, 3–9
 _SIMPLE_READ built-in function, 10–13
 _SIMPLE_WRITE built-in function, 10–14
sizeof keyword, 6–10
_SKPC built-in function, 10–14
Source code

Source code (Cont.)
 output listing section, 2–19
Source code listing
 general description, 2–19
source command (dbx), 3–11
Source files
 creating (vi), 2–3
 editing (vi), 2–3
Source programs
 migrating
 See Portability concerns
_SPANC built-in function, 10–15
Standard argument
 vcc command (–V option), 2–16
standard pragma, 9–11
Statements, 5–1 to 5–9, D–4
 break, 5–8
 case, 5–4
 compound, 5–2
 conditional, 5–3 to 5–6
 continue, 5–8
 control flow, 5–1 to 5–2
 default, 5–4
 do, 5–7
 expressions as, 5–2
 for, 5–6
 goto, 5–2
 if, 5–3
 interrupting, 5–8 to 5–9
 labels, 5–2
 looping, 5–6 to 5–8
 null, 5–1
 return, 5–9
 switch, 5–4
 while, 5–7
static specifier, 8–9
Static storage class, 8–9
status command (dbx), 3–11
step command (dbx), 3–11
 multiple steps
 command-line retention, 3–3
stepli command (dbx), 3–14
stop command (dbx), 3–12
stopi command (dbx), 3–14
Storage
 modifiers, 8–16
Storage allocation, 8–7
 for program sections, 8–7
 lifetime of variables, 8–7
 location of, 8–7
 registers, 8–7
 run-time stack, 8–7
Storage classes
 defined, 8–1
 external, 8–10 to 8–11
 definitions and declarations, 8–10
 global, 8–11
 in determining scope, 8–1
 internal, 8–7
 auto specifier, 8–8
 register specifier, 8–9
 introduction to, 4–3
 list of, 8–3
 modifiers, 8–17 to 8–19
 _align, 8–18
 const, 8–16
 introduced, 8–3

- Storage classes
 - modifiers (Cont.)
 - noshare, 8-18
 - readonly, 8-18
 - volatile, 8-17
 - order of keywords in declarations, 8-2
 - specifiers
 - auto, 8-4, 8-8
 - extern, 8-4, 8-10, 8-11
 - globaldef, 8-11
 - globalref, 8-11
 - globalvalue, 8-14
 - list of, 8-4
 - register, 8-9
 - static, 8-9
- Storage-class modifiers, D-3
- Storage-class specifiers, D-3
- Storage map
 - output listing section, 2-20 to 2-23
- String data type
 - See also Arrays
 - declaration of, 7-14
- Strings
 - assignment, 4-15, 4-16
 - concatenation of, 7-14
 - continuation, 7-14
- Structures, 7-15
 - alignment, 9-11
 - bit fields, 7-22
 - declaration of, 7-15, 7-16 to 7-17
 - forward referencing, 7-17
 - initialization of, 7-18
 - introduction to, 4-15
 - members of
 - references to, 6-4, 7-17 to 7-18
 - padding, 1-11
 - variant aggregates, 7-20
- Substitution
 - macro, 9-3 to 9-5
 - in #include directives, 9-8
- Subtraction operator, 6-10
- _SVPCTX built-in function, 10-15
- switch** statement, 5-4, 5-6
 - declarations inside of, 5-5
 - introduction to, 4-8
- Symbolic constants, 7-1
- Symbol table
 - created by compiler, 2-4
- Syntax
 - main function, 4-28

T

- \t (horizontal tab), 7-5
- Tags
 - vacuous declarations, 7-17
- Tape files
 - See Files
- tar utility (DEC/Shell)
 - using to handle magnetic tape, C-2
- Text editor (vi), 2-3
 - See also Editor
 - environment, 2-3
- __TIME__ predefined macro, 10-3
- t option
 - vcc command option, 2-8

- Traceback mechanism
 - effect of debug argument (-V option), 2-11
- trace command (dbx), 3-12
- tracei command (dbx), 3-14
- typedef** keyword, 7-23
- Type specifiers, D-2

U

- ULT\$COMM, A-6
- ULT\$DATA, A-6
- ULT\$TEXT, A-6
- ULTRIX vcc command, D-1
- unalias command (dbx), 3-12
- Unary expressions
 - address of, 6-8
 - cast, 6-9
 - increment and decrement, 6-7
 - indirection, 6-8
 - negation, 6-7
 - one's complement, 6-9
 - sizeof, 6-10
- Unary operators, 7-8
 - precedence of, 6-6
- #undef** directive, 2-16, 9-5
- Unions, 7-15
 - declaration of, 7-15, 7-16
 - introduction to, 4-15
 - members of
 - references to, 6-4
 - variant aggregates, 7-20
- Unresolved references, 2-18
- unset command (dbx), 3-12
- Unsigned conversion, 6-16
- U option
 - vcc command option, 2-8
- up command (dbx), 3-13
- use command (dbx), 3-13
- User-defined functions
 - See Functions
- /usr/lib/fortmsgfile, C-4
- Utilities
 - lint, 4-32

V

- \v (vertical tab), 7-5
- Vacuous tag declarations, 1-7, 7-17
- Values
 - defined, 7-1
 - of constants, 7-1
 - of variables, 7-1
- Variables
 - aggregate, 7-2
 - character, 7-3
 - declarations
 - format of, 7-2
 - introduction to, 4-3
 - declared in overlapping blocks, 4-31
 - identifiers, 4-29
 - objects of, 7-1
 - scalar, 7-2
 - values of, 7-1
- variant_struct**, 7-20
- variant_union**, 7-20
- VAX C language

VAX C language (Cont.)

See also Portability concerns
introduction to, 4-2
keywords, 4-29

introduction to, 4-4
list of operators, 6-4
program structure, 4-18
introduction to, 4-2

VAX C tutorial, 4-2 to 4-18

See also Statements

addresses, 4-12
aggregates, 4-14 to 4-18
arguments, 4-3
arrays, 4-14
blocks, 4-7
break statement, 4-9
case sensitivity, 4-4
character strings, 4-14
comments, 4-3
compiling and linking, 4-6
compound statement, 4-7
conditional execution, 4-7, 4-8, 4-9, 4-10
data types, 4-3
definition files, 4-5
do statement, 4-9
equality operator, 4-7
for statement, 4-10
function body, 4-3
functions, 4-3
if statement, 4-7
input/output, 4-4
isupper macro, 4-8
keywords, 4-4
language keywords, 4-4
linking against libraries, 4-4
loop incrementing, 4-11
loops, 4-9
macros, 4-8
newline character, 4-5
OR operator, 4-7
parameters, 4-3
pointers, 4-12
preprocessor directives, 4-8
program flow, 4-6
return statement, 4-3
scalars, 4-14 to 4-18
storage classes, 4-3
structures and unions, 4-15 to 4-18
switch statement, 4-8, 4-9
_tolower macro, 4-8
ULTRIX, 4-6
ULTRIX file extensions, 4-6
ULTRIX file names, 4-6
values, 4-12
variable declarations, 4-3
void function, 4-3
while expression, 4-10
white space, 4-3

VAX C/VMS source programs

portability concerns, 1-1

vcc command, D-1

differences from CC command, C-3 to C-5
input and output files, C-3
search paths used in, C-4

vcc command (Shell), 2-4 to 2-17

cross_reference argument (-V option), 2-10
debug argument (-V option), 2-10 to 2-11

vcc command (Shell) (Cont.)

define argument (-V option), 2-11 to 2-12
format, 2-4
g_float argument (-V option), 2-12
list argument (-V option), 2-13
machine_code argument (-V option), 2-14
object argument (-V option), 2-14
optimize argument (-V option), 2-14
show argument (-V option), 2-15
specifying files in, 2-5, 2-6, 2-7
standard argument (-V option), 2-16
undefine argument (-V option), 2-16
-V option summary, 2-10
warnings argument (-V option), 2-16

vcc command line

options affecting debugging, 3-2

vcc command options

options affecting debugging, 3-2

vi editor

See Text editor

void, 4-23 to 4-24

void keyword, 7-22

void keyword

pointers, 7-9

void pointers, 7-9

volatile modifier, 8-17

-v option

vcc command option, 2-8

-V option

vcc command option, 2-8

-V option (vcc command)

summary, 2-10

W

warnings argument

vcc command (-V option), 2-16

whatis command (dbx), 3-13

where command (dbx), 3-13

whereis command (dbx), 3-13

which command (dbx), 3-13

while statement, 5-7

White space, 4-32

-w option

vcc command option, 2-8

_WRITE_GPR built-in function, 10-15

X

XOR bitwise operator, 6-12

Y

-Y option

vcc command option, 2-8

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	_____	Local Digital subsidiary or approved distributor
Internal*	_____	SSB Order Processing - WMO/E15 <i>or</i> Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to VAX C for ULTRIX
AA-ME83B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What would you like to see more/less of? _____

What do you like best about this manual? _____

What do you like least about this manual? _____

Please list errors you have found in this manual:

Page	Description
------	-------------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear - Fold Here and Tape

digital™



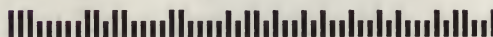
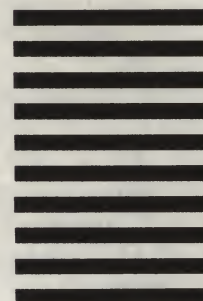
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZK03-2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062-9987



Do Not Tear - Fold Here

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Guide to VAX C for ULTRIX
AA-ME83B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What would you like to see more/less of? _____

What do you like best about this manual? _____

What do you like least about this manual? _____

Please list errors you have found in this manual:

Page	Description
------	-------------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear - Fold Here and Tape

digital™



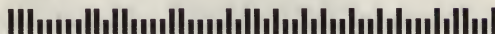
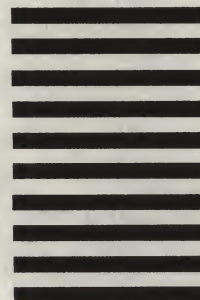
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062-9987



Do Not Tear - Fold Here

Cut
Along
Dotted
Line

digital